DATABASE MANAGEMENT SYSTEMS
Volume-1: Study Material with Classroom Practice Questions
2. ER and Relational Model

01. Ans: (b)
Sol: Derived attribute is an attribute that derives its value from one or more attributes.

02. Ans: (b)
Sol:

```
Employee ----> Manage ----> Department
```

03. Ans: (a)
Sol: As every specialized entity is subset of generalized entity, then the deletion of generalized entity requires the deletion of specialized entity.

04. Ans: (c)
Sol:
- Composite attribute is an attribute which is composed of other attributes.
- Multi valued attribute represented with double ellipse.
- Derived attributes represented with dotted ellipse

05. Ans: (b)
Sol:

```
Professor ----> Teaches ----> Course
```

06. Ans: (a)

```
E1 ----> M:1 R ----> E2
```

07. Ans:
Sol:

```
Professor ----> Teaches ----> course

Professor ----> Teaches ----> course

Professor ----> Teaches ----> course

Professor ----> Teaches ----> course
```
08. Ans: (c)
Sol: Set of attributes which contains a candidate key is said to be a super key.

09. Ans: (b)
Sol: All the values present in Foreign key must present in primary key of the referenced relation.

10. Ans: (c)
Sol: It violates referential integrity constraint as it is updating in foreign key but not in primary key.

11. Ans: (c)
Sol: When parent is update, it requires child table to be updated simultaneously

12. Ans: (c)
Sol: 1. On removal of row (2,4), row (5,2) and (7,2) must also be deleted as they depend on value

13. Ans: 0
Sol: When <3, 8> is deleted, its related tuples in T2 is (8, 3) and 3 is to set null. Hence the number of additional tuples to delete is 0

14. Ans: (a)
Sol: As the key constraint from professor, the maximum number of tuples possible in Teaches is number of tuples in professor.

15. Ans: (a)
Sol: 

16. Ans: (a)
Sol: As C, A and B in total participation with R1 and R3 and there is key constraint, all these are represented with one relation and R2 is represented separately one relation.

17. Ans: (b)
Sol: There are some tuples of course may not participate with any tuple of professor, then cid is a key for the relation.

18. Ans: (a)
Sol: (AR1B) will be one table as there is total participation and key constraint. (CR2) will be the second table as there is a key constraint.
19. Ans: (b)  
Sol: As we get key and participation constraint from course to registration, therefore the number of tuples in registration will be equal to the tuples in the course table.

20. Ans: (b)  
Sol: Strong entities $E_1$ and $E_2$ are represented as separate tables, in addition to that many to many relationship ($R_2$) must be converted as separate table by having primary key of $E_1$ and $E_2$ as foreign key. One to many relationship must be transferred to ‘many’ side table by having primary key of one side as foreign key. Hence we will have minimum of 3 tables.

21. Ans: (b)  
Sol: Strong entities E1 and E2 are converted as separate tables. Since A23 is a multi valued attribute it should also be converted as separate table. Relationship R is transferred to ‘m’ side (E2).

22. Ans: 3  
Sol: E-R model is

```
Employee  manages  Department  sponsors  Project
```

The minimum number of relations in relational model is 3.  
1. (Employee, manages)  
2. Department  
3. (Project, sponsors)

23. Ans: (c)  
Sol: 

```
Person  has  Loan
```

As there is a key constraint from Loan, Relationship can be merged with Loan entity set.

24. Ans: (b)  
Sol: M, P are strong entities hence they must be represented by separate tables. M table is modified to include primary key of P side (i.e P1). N is weak entity, and it is modified to include primary key of P (i.e P1).

25. Ans: (a)  
Sol: M and P are strong entities hence they must be represented as separate tables. To include R1, M table is modified to accommodate primary key of P side (i.e P1) as foreign key. N is weak entity, so modify N to accommodate primary key of P (i.e P1) as foreign key. Therefore tables are (M1, M2, M3, P1), (P1, P2), (N1, N2, P1). So correct answer is (M1, M2, M3, P1).

26. Ans: (b)  
Sol:  
- An Entity type is represented with one relation.  
- Key attribute becomes primary key for the relation  
- Composite attribute is represented with set of simple attributes  
- Weak entity is represented always as a child table
3. Functional Dependencies

01. Ans: (d)
   Sol: As ‘BC’ is key BC→A is satisfied

02. Ans: (b)
   Sol: Based on the table values given in query and guidelines below, answer is b. YZ is having unique combination, and Y is also having unique values. Hence YZ→X, Y→Z are possible.

03. Ans: (b)
   Sol: A functional dependency X→Y is satisfied if any two tuples agree on X then they must also agree on Y.

04. Ans: (c)
   Sol: AF⁺ = AFDE not ACDEFG as given.

05. Ans: (e)
   Sol: A functional dependency X→Y is said to be trivial iff Y⊆X.

06. Ans: (b)
   Sol: CD⁺ from functional dependencies (FDs) = CDEAB, it includes RHS attributes AC so it can be derived from FDs BD⁺ from functional dependencies (FDs) = BD only, RHS attributes CD are not included in the closure hence it cannot be derived BC⁺ from functional dependencies (FDs) = BCD EA, it includes RHS attributes CD, so it can be derived from FDs AC⁺ from functional dependencies (FDs) = ACBDE, it includes RHS attributes BC so it can be derived from FDs

07. Ans: (e)
   Sol: AC⁺ contains I then AC→I dependency is possible.

08. Ans: 24

09. Ans: (b)
   Sol: As ‘K’ is independent attribute, key is ABDK.

10. Ans: (d)
    Sol: ABD⁺ = A, B, C, D, E.

11. Ans: (b)
    Sol: ACEH⁺ contains all the attributes of R.

12. Ans: (d)
    Sol: Closure of AEH⁺ = BEH⁺ = DEH⁺ = A, B, C, D, E, H. If any closure includes all attributes of a table then it can become candidate key of the table. Closure of AEH, BEH, DEH includes all attributes of table. Hence they are candidate keys.

13. Ans: (b)
    Sol: Set of attributes which contains a candidate key is called super key.
14. Ans: 3  
Sol: The candidate keys are  
F  
AB  
CB  

15.  
Sol: CK: ACD, BCD, ECD.  

16. Ans: 6  
Sol: AB, AD, EB, ED, CB, CD.  

17. Ans: 2  
Sol: D, AH  

18. Ans: (b)  
Sol: The number of super keys are A, B, C, AB, AC, BC, ABC.  

19. Ans: (a)  
Sol: D→E of F is not covered by G.  

20. Ans: (c)  
Sol: D→C in set2 and C→D in set1 not covered by each other.  

21. Ans: (d)  
Sol: AB→C, A→BC both can be determined from remaining set of FD’s.  

22. Ans: 5  
Sol: AC → D can be eliminated, it can be derived from A → B and CB → D using augmentation and transitive rule.  
A → B ⇒ AC → BC  
⇒ AC → D  

And remaining FD’s are not possible to eliminate  
∴ 5 FD’s are there in minimal cover.  

23. Ans: (c)  
Sol: The minimal set of F is A→B, B→C and AB→C is redundant.  

24. Ans: (a)  
Sol: As V → W, delete W from VW → X results in V → X  
As V → X, delete X from Y → VX results in Y → V  
The irreducible set is  
V → W  
V → X  
Y → V  
Y → Z  

25.  
Sol:  
\[ \begin{align*}  
AD & \rightarrow CF 
\end{align*} \]  
\[ \begin{align*}  
C & \rightarrow B 
\end{align*} \]  
\[ \begin{align*}  
B & \rightarrow E 
\end{align*} \]  
\{ \text{Canonical set} \}  

26.  
Sol:  
\[ \begin{align*}  
A & \rightarrow BC  
AE & \rightarrow H 
C & \rightarrow D  
D & \rightarrow G  
E & \rightarrow F 
\end{align*} \]  
\{ \text{Minimal set} \}
01. Sol: 1. C.K = BD, Lossy, Dependency preserving
   2. C.K = AB, CB, Loss-less, Not Dependency preserving
   3. C.K = A, C, Loss-less, Dependency preserving
   4. C.K = A, Loss-less, Not Dependency preserving
   5. C.K = A, Lossy, Not Dependency preserving

02. Ex: F: {AB→C, A→D}

03. Ex: F: {AB→C, C→D}

04. Sol: R is in 1NF :: decompose to 2NF
   A⁺ = {A, D, E, I, J}; R₁ = 2NF
   B⁺ = {B, F, G, H}; R₂ = 2NF

   {A, B, C}; R₃ BCNF
   Then decompose into 2NF
   R₁ (ADEIJ)
   R₂ (BFGH)
   R₃ (ABC)
   3NF also in BCNF
   R₄ (DIJ)
   R₅ (AED)

05. Sol: Candidate key: AC
   A⁺ = (ABE) R₁, C⁺ = (CD) R₂
   (ACF) R₃

06. Sol: (1) C → D
   C → A
   B → C
   C.K: B, 2NF but not 3NF
   (2) 2NF but not 3NF as no partial dependency CK: BD.
   (3) R is in 3NF but not in BCNF
   (4) C.K = A
   (5) Candidate Keys = AB, CD, BC, AD
   R is in 3NF but not in BCNF.

07. Ans: (d)
   Sol: 3NF requires for a non trivial FD of the form X → A then X is super key (or) A is prime attribute.

08. Ans: (a)
   Sol: As given client id and order id together is a key and it is possible to determine Firstname, Lastname of a client using his client id, then we have the dependency clientid → Firstname, Lastname which is a partial functional dependency. Hence the relation is in 1 NF.
09. Ans: (b)
Sol: (Volume, Number) → Year is a partial functional dependency. So, the given relation is in 1 NF but not in 2 NF.

10. Ans: (c)
Sol: R is in 1NF as A→FC and B→E are partial dependencies

11. Ans: (d)
Sol: Relation R₁ satisfies A→B, B→C and C→AB dependencies and all the determinants are super keys. Hence the relation is in BCNF.

12. Ans: (c)
Sol: In CD→B; B is prime attribute.

13. Ans: (b)
Sol: rollno, courseid is superkey in rollno, courseid→email, rollno is prime attribute in email→rollno.

5. Structured Query Language (SQL)

01. Ans: (b)
Sol: The result of the query is

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

02. Ans: (c)
Sol: \[
\text{avg (marks)} = \frac{(10 + 0 + 30 + 0)}{4} = 10
\]

03. Ans: (b)
Sol: Select clause contains either aggregate function or the attributes that appear in group by clause.

04. Ans: (c)
Sol: \[
\text{sum (rating)/count(0)} \text{ is smaller value than } \text{avg(rating)}.
\]

05. Ans: (a)
Sol: If A→B does not hold, we expect some output.

06. Ans: (c)
Sol: Union operator eliminates the duplicates.

07. Ans: (c)
Sol: The result of the query is

<table>
<thead>
<tr>
<th>e1. empno</th>
<th>e2. empno</th>
</tr>
</thead>
<tbody>
<tr>
<td>326</td>
<td>321</td>
</tr>
<tr>
<td>350</td>
<td>323</td>
</tr>
<tr>
<td>351</td>
<td>323</td>
</tr>
<tr>
<td>351</td>
<td>350</td>
</tr>
</tbody>
</table>

08. 2
Sol: It returns two rows.

<table>
<thead>
<tr>
<th>Student – Name</th>
<th>Sum(P.Marks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raj</td>
<td>4</td>
</tr>
<tr>
<td>Rohit</td>
<td>2</td>
</tr>
</tbody>
</table>
09. Ans: (c)
Sol: 

<table>
<thead>
<tr>
<th></th>
<th>R₁</th>
<th>R₂</th>
<th>R₁⋈R₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>4</td>
<td>9</td>
</tr>
</tbody>
</table>

10. Ans: 8
Sol: Full outer join of R and S will give T relation. Here NULL entries are taken for R and S to include all missing instances of common attribute A while joining R and S.

11. Ans: (d)
Sol: Inner query retrieves average salary of all employees. Outer query computes average salary of all male employees. Therefore it retrieves department names where average salary of male employees is more than the average salary in the company.

12. Ans: (b)
Sol: The condition is B>any (1,2,1,3,2,4) and the output will be 4.

13. Ans: (d)
Sol: Finds rating and average age of each rating of those sailors whose age is >= 18 and there is atleast two sailors for each rating.

<table>
<thead>
<tr>
<th>S.rating</th>
<th>avg(S.age)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
</tr>
</tbody>
</table>

14. Ans: (a)
Sol: all(empty) returns true always.

15. Ans: (c)
Sol: Inner Query finds the number of guests trained by JULIO

16. Ans: 7
Sol: The output of the query is

<table>
<thead>
<tr>
<th>ta.player</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klose</td>
</tr>
<tr>
<td>Ronaldo</td>
</tr>
<tr>
<td>G muller</td>
</tr>
<tr>
<td>Fontaine</td>
</tr>
<tr>
<td>Pele</td>
</tr>
<tr>
<td>Klismann</td>
</tr>
<tr>
<td>Kocsis</td>
</tr>
</tbody>
</table>

17. Ans: (d)
Sol:

```
<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Age</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>A</td>
<td>16</td>
</tr>
<tr>
<td>B</td>
<td>17</td>
</tr>
<tr>
<td>C</td>
<td>15</td>
</tr>
<tr>
<td>D</td>
<td>21</td>
</tr>
<tr>
<td>E</td>
<td>22</td>
</tr>
</tbody>
</table>
```

When the query executes on the above table of data returns A,D,E in the output, those are students with higher rank than all students with age <18.
**Exercise - 01**

01. **Sol:** Display the details of all hotels, located in London.

02. **Sol:** Display name & address of all guests in ascending order of their name living in London.

03. **Sol:** Display the details of all rooms of price below 40 & type of the room is either ‘D’ or ‘F’ in ascending order of their prices.

04. **Sol:** Display all booking details whose check out time not specified.

05. **Sol:** Finds total price of all rooms of type ‘D’.

06. **Sol:** Find the number of guests who have a booking on 1st August or in the month of August.

07. **Sol:** List the type & price of all rooms in Grosvenor hotel.

08. **Sol:** List all the guest details who have a booking for the current date in “Grosvenor Hotel”

09. **Sol:** List the details of all rooms at the ‘Grosvenor hotel’ including the name of the guest staying in the room, if the room is occupied.

10. **Sol:** Finds total price of all rooms in ‘Grosvenor hotel’ if the room is currently occupied.

11. **Sol:** Find number of rooms in each hotel located in London.

12. **Sol:** Find the maximum number of rooms booked in London of same type.

   |   |   |
---|---|---|
**A** | 100 | (Room Type) |
**B** | 200 |   |
**C** | 150 |   |
**D** | 75  |   |

13. **Sol:** Find total price of all rooms, that are not currently booked in each hotel.

Maximum 200. It displays total no. of bookings for the most popular room in city of London.
6. Relational Algebra & Calculus

01. Ans: (b)
Sol: As list1 \subseteq list2 the result will be equal to
\[ \prod_{\text{list1}} R \]

02. Ans: (d)
Sol: Relational Algebra eliminate duplicates always.

03. Ans: (c)
Sol: If \((X=Y) \subseteq Z\) then the expressions to be true.

04. Ans: (c)
Sol: In the order of evaluation, first we perform selection, then we perform perfection.

05. Ans: (a)
Sol: \( \Pi_B(r_1) - \Pi_C(r_2) = \phi \) is always true. Because ‘B’ is foreign key referencing ‘C’, so ‘C’ must be a primary key, ‘B’ cannot have a value that is not available in ‘C’. Hence operation \( \Pi_B(r_1) - \Pi_C(r_2) \) is always \( \phi \).

06. Ans: (a)
Sol: Common column between tables ‘R’ and ‘S’ is attribute B. In table ‘R’ B is primary key (\( B \rightarrow A \), \( A \rightarrow C \)). In table ‘S’ B is foreign key so join is performed on attribute B. Therefore maximum tuples possible in the output is equal to rows in Table S (as it has less number of rows, provided B values are not repeated in table ‘S’).

07. Ans: (b)
Sol: bal < 0 filter rows from account \( \infty \) depositor from which we can operate on few rows to filter b city = “Agra”.

08. Ans: (d)
Sol: Minus operator indicates rows available in LHS table but not in RHS table. In this expression, LHS table produces all female students, RHS table consists students with less marks hence it produces names of all girl students with more marks than all the boy students.

09. Ans: (a)
Sol: Apply first cross product then apply filter. Cross product yields 10 rows, then you play filter \( A.ID>40 \) or \( C.ID<15 \) produces 7 rows.

\[ A \bowtie B \Rightarrow A.ID \]
\[ 12, 15, 25, 98, 99 \]
\[ x \]
\[ \frac{12}{15} \]
\[ 15, 25, 98, 99 \]
\[ \frac{25}{99} \]
\[ 99, 10 \]
\[ \frac{99}{99} \]

Result contains 7 rows
11. Ans: 4
Sol: The output of $T_1$ is: `courseName`
    CA
    CB
    CC
the output of $T_2$ is: `StudentName`
    SA
    SC
    SD
    SF

12. Ans: (c)
Sol: $P$. duration = 3 months selects all projects of
duration 3 months
$T$. pname = $P$.name selects project names in
the output.

13. Ans: (a)
Sol: SQL, Relational algebra, tuple relational
calculus and Domain relational calculus all
is representing the same. i.e., all these
expressions representing to find the distinct
names of all students who score more than
90% in the course numbered 107.

14. Ans: (d)
Sol: Minus operator indicates, rows available in
LHS table but not in RHS table. In this
expression, LHS table produces all female
students, RHS table consists students with
less marks hence it produces names of all
girl students with more marks than all the
boy students.

15. Sol: Result of given query

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>b</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>d</td>
<td>6</td>
</tr>
</tbody>
</table>

16. Ans: 2
Sol: Relational calculus eliminate the duplicates.
\{T | \$B \in \text{Book} \land (T.\text{Title} = B.\text{Title})\}

17. Sol:
\[
\begin{align*}
(a) & \quad \{T | \exists M \in \text{Manager} \\
& \quad \land (M.\text{Manager\_name} = \text{'Jones'} \land \\
& \quad \land (T.\text{Person\_name} = M.\text{Person\_name})
\}
\end{align*}
\]

(b) \[
\begin{align*}
\{T | \exists E \in \text{Employee} \\
& \quad \land (T.\text{City} = E.\text{City}) \land \exists M \in \text{Manager} \\
& \quad \land (M.\text{Manager\_name} = \text{'Jones'} \land \\
& \quad \land (E.\text{Person\_name} = M.\text{Person\_name})
\}
\end{align*}
\]

(c) \[
\begin{align*}
\{T | \exists E \in \text{Employee} \\
& \quad \land (E.\text{Person\_name} = \text{'Jones'} \land \\
& \quad \land (T.\text{Manager\_name} = E.\text{Manager\_name})
\}
\end{align*}
\]
7. Transactions & Concurrency Control

01. Ans: (d)
Sol: Transaction $T_3$ perform read on $A$, which is updated by $T_1$ and committed before $T_1$ does.

02. Ans: (a)
Sol: As $R_3(x)$ is dirty operations which read $W_1(x)$ and is committed before $T_1$. Hence schedule is non-recoverable.

03. Ans: (c)
Sol: A recoverable schedule is one where for each pair of transactions $T_i$ and $T_j$ such that $T_j$ reads a data item previously written by $T_i$, the commit operation of $T_i$ appear before the read operation of $T_j$.

04. Ans: (c)
Sol: A schedule is said to be strict if a value written by a transaction $T$ is to be read or written by another transaction until either $T$ commits or aborts.

05. Ans: (d)
Sol: A recoverable schedule is one where for each pair of transactions $T_i$ and $T_j$ such that $T_j$ reads a data item previously written by $T_i$, the commit operation of $T_i$ appear before the read operation of $T_j$.

A schedule is said to be strict if a value written by a transaction $T$ is to be read or written by another transaction until either $T$ commits or aborts.

06. Ans: (b)
Sol: The number of serial schedules are 2
The number of concurrent schedules are $\frac{(5+3)!}{5!*3!} = 56$
Then, the total number of non serial schedules are $= (\text{number of concurrent schedules} - \text{number of serial schedules})$
$= 56 - 2 = 54$

07. Ans: (b)
Sol: Every cascadeless schedule is recoverable but need not vice versa.

08. Ans: (a)
Sol: $R_1(X), R_1(Y), R_3(Y), W_2(Y), W_1(X), W_3(X), R_2(X), W_2(X)$.
$R_3(Y), W_2(Y)$ are conflicting hence serializability says $T_3$ must be before $T_2$ in any serial schedule.
$W_1(X), W_3(X)$ are conflicting hence serializability says $T_1$ must be before $T_3$ in any serial schedule.
$W_3(X), R_2(X)$ are conflicting hence serializability says $T_3$ must be before $T_2$ in any serial schedule.
The serial schedule $T_1, T_3, T_2$ is satisfying all the above conditions. Directed graph for the non-serial schedule will also give same sequence.
09. Ans: (a)  
Sol:  

<table>
<thead>
<tr>
<th></th>
<th>T₁</th>
<th>T₂</th>
<th>T₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(x)</td>
<td>R(y)</td>
<td>R(y)</td>
<td>R(x)</td>
</tr>
<tr>
<td>R(z)</td>
<td>R(x)</td>
<td>R(z)</td>
<td>W(y)</td>
</tr>
<tr>
<td>W(x)</td>
<td>W(x)</td>
<td>W(z)</td>
<td></td>
</tr>
</tbody>
</table>

Precedence graph

S₁ is conflict serializable to T₂ → T₃ → T₁

S₂:  

<table>
<thead>
<tr>
<th></th>
<th>T₁</th>
<th>T₂</th>
<th>T₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(x)</td>
<td>R(y)</td>
<td>R(y)</td>
<td>R(x)</td>
</tr>
<tr>
<td>R(z)</td>
<td>R(x)</td>
<td>R(z)</td>
<td>W(y)</td>
</tr>
<tr>
<td>W(x)</td>
<td>W(x)</td>
<td>W(z)</td>
<td></td>
</tr>
</tbody>
</table>

Precedence graph

S₂ is not conflict serializable

10. Ans: (d)  
Sol:  

Option (A)

Option (B)

Option (C)

Option (D)

Conflict Serializable

11. Ans: (e)  
Sol: Precedence graph is

Precedence graph
12. Ans: (d)
Sol: S1 and S2 are conflict equivalent to serial schedule T2, T3, T1.
S3 is not conflict equivalent as 2RA, 3WA (T2<T3) and 3WA, 2WA (T3<T2) are the conflict operations. There is no serial schedule that satisfies both T2<T3 and T3<T2.

13. Ans: (b)
Sol: As there is a cycle in precedence graph, the schedule is not conflict serializable, but satisfying view rules.

14.
Sol: (a) Not Conflict Serializable,
Not View Serializable,
Recoverable, Avoids Cascading aborts,
Not strict.
(b) Not Conflict Serializable,
Not View Serializable,
Not strict,
Recoverable, cascading aborts
(c) Conflict Serializable,
View serializable,
Serializable,
Recoverable,
Avoids cascading aborts,
Not strict
(d) Not Conflict Serializable,
Views serializable through Thomas write rule, Serializable,
Recoverable,
Avoids cascading aborts,
Not strict
(e) Conflict Serializable,
Not View Serializable,
Sequentializable,
Recoverable,
Avoids cascading aborts,
Not strict
(f) Conflict Serializable,
View serializable,
Sequentializable,
Serializable,
Recoverable,
Avoids cascading aborts,
Not strict

15. Ans: (c)

16. Ans: (c)
Sol: An older transaction requesting a data item held by an younger Tx need to wait.
17. Ans: (b)
Sol: Construct the wait-for-graph, contains edges from T1→T2, T2→T3, and T4→T2 and there is no circular wait, hence no deadlock.

18. Ans: (d)
Sol: W1(a), R2(a) and W2(b), R1(b) are conflicting, going by the principles of serializability it is not serializable and cannot occur in 2PL.
09. Ans: (d)  
Sol: Deleting ‘10’ from internal node requires 10 to be replaced with copy of 13.

10. Ans: (a)  
Sol: Insert 15

11. Ans: (a)  
Sol: If we remove K50 in index node, there is no change in height as still root, index and leaf nodes exist.
Leaf nodes are now, after Delete 50, the B⁺ is:

Hence Root now consists: 20
(i) is true, (ii) is true but (iii) is not true